

Should You Migrate to Angular?



AN URUIT CASE STUDY

UruIT

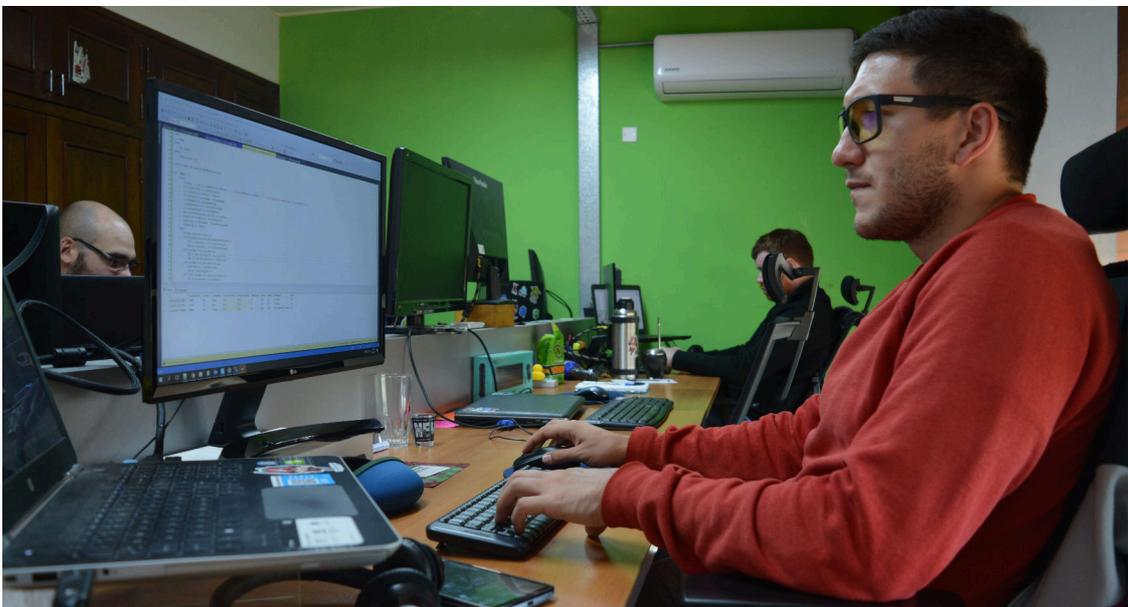


Are you considering migrating your application from **AngularJS** to **Angular**?

In this case study, we'll describe a project in which we successfully managed the migration and share our recommendations based on what we have learned.

Background

At UruIT, one of our biggest partnerships is with a leading provider of software products for the commercial real estate industry in the US, serving 50 markets with over 2 million properties registered throughout the country. Over the last two years, we have worked together to create the most comprehensive listing and property information platform, composed of different satellite tools (mostly web apps), including reporting tools, map-based searches, an email platform, file folders, and location analytics tools among several others.



We are organized in different Scrum teams at UruIT, taking a co-development approach, for a total of 20 senior developers focused

on implementing the front end of different sets of features for this client with delivery locations in Uruguay and Colombia as well as an in-house team located inside the US.

In the beginning of the project, we started coding in early versions of AngularJS, but the migration to Angular (formerly known as Angular 2) was our final goal, and a year later, we began the process.

Here's how we did it.

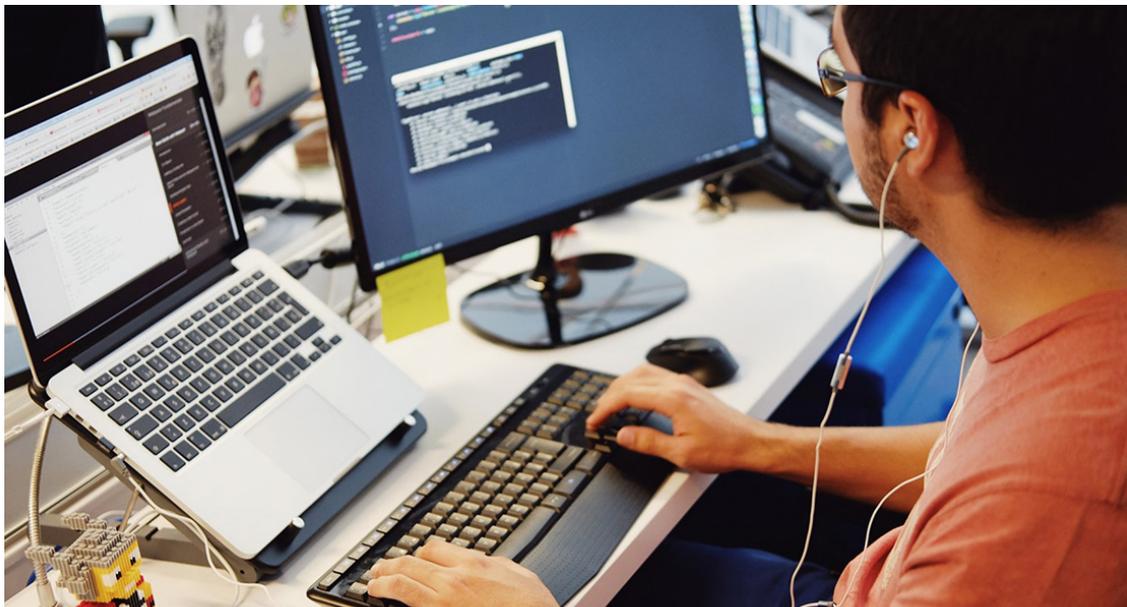
Method



Since we were working with AngularJS, before starting the migration itself, we took some initial steps that facilitated the journey:

1. At the outset, we developed in both AngularJS and Typescript, because we believed it would bring more benefits and also help us with the future migration. **Learn more here!**
2. We lined up all available training, including an in-company training. (Our best friend during the process was the Angular migration official **documentation**. It will most likely be yours too!)

3. Some of the tech leads built a POC for the very first Angular feature and gradually the rest of the Dev team took part in the process. We realized there's nothing like learning by doing and with our incremental approach, we had a very productive work rhythm. The software codebase was huge and the release plan very challenging. Interrupting its development in order to migrate wasn't going to be an option. So, our decision was to follow the product evolution roadmap and work in the migration in parallel, taking the Vertical slicing approach. A gradual evolution and behind the scenes, we managed to code new features directly in Angular, while upgrading selected features of the old part of the software.



Today, around 80% of the application is written in Angular and both of the frameworks co-exist, since there are some parts that were simply too complex to migrate.

Benefits

- ✔ An internal benefit, the challenge of developing products and leading a migration at the same time was met by our developers with enthusiasm and determination. Working with updated technology motivated them, since a brand new framework brought **new opportunities for learning and improving**.
- ✔ Product-wise, the migration proved beneficial since Angular change detection is a lot faster than AngularJS and it allows customizations when required. Also, it's easier to spot template compilation issues at the time of debugging, helping us to **work more efficiently and accurately**.
- ✔ These days, AngularJS third party libraries are being deprecated and Angular libraries are gaining a lot more traction. This meant that continuing to develop in AngularJS could have limited the growth of the products. Migrating allowed our client to **think in the long-term**.

Our Recommendation

If you are still indecisive, we believe that it is, in most cases, worth it to migrate. Even though the migration that we carried out from AngularJS to Angular was slightly intimidating, future version upgrades are not going to be that complicated. In fact, migrating from version 2 to version 4, for example, is straightforward and almost no change in the application is required.

If you need a second opinion, **Stephen Fluin, Google Developer Advocate for Angular**, lists other benefits:



“You would want to migrate if:

- 1. You want to begin server-rendering your application*
- 2. You are looking for better runtime performance*
- 3. You want to take advantage of powerful new capabilities like TypeScript, ES2016 syntax, code splitting, PWAs, etc.*
- 4. 3rd party tools you rely on are thinking about moving away from AngularJS and towards Angular.*
- 5. Your developers want to write Angular”*

Checklist for an incredible migration

It's possible to be prepared and you can always find solutions for any migration issues within the community. These are our tips for you:

- ✓ Read and consult **Angular's documentation** along the way.
- ✓ Collaborate with the community, which is always finding and sharing tips. One example is the **ng doc**, a unofficial documentation that complements that of Angular.
- ✓ If you plan to follow an incremental migration path, you have to be prepared for code duplication, since you will build the new version but maintain the old one to be sure everything works perfectly. But, in no time, you will have everything updated, so don't give up!
- ✓ During the migration, the final javascript file will be heavier, so be careful not to make your users face overhead by having two frameworks working simultaneously. If it's only a web application, the problem is not a big deal, but for mobile apps it can be risky.

Final Thoughts



Although we generally endorse migrating to Angular for the reasons listed above, if you have a product already developed in AngularJS and everything works great, in certain situations, the benefits might not outweigh the required effort. If this is true for you, you can always develop complementary products or functionalities in Angular as you get to know this incredible evolution.

There are different ways to conduct a migration from AngularJS to Angular. We told you the way that worked the best for us, but you may want to evaluate all of the different options available.

Our advice: if you have a small or medium size app, it may be possible to stop development and migrate everything before continuing. That way, you will avoid code duplication and other challenges. However, for more complex apps, it can be tough. We recommend the Vertical approach to get ahead of problems and doing the full migration of each process before going on to another.

Further Resources



- [NG-Conf YouTube channel](#)
- [Upgrading Enterprise Angular Applications](#)
- [Angular 2 Migration Path](#)
- [What's new in Angular 4](#)
- [Advantages of Angular 2](#)

If you have any doubts about the migration or wants any help with it, let's get connected!

hello@uruit.com
@UruIT